

Windows Programming

Visual C++ MFC Programming

Lecture 04

김예진

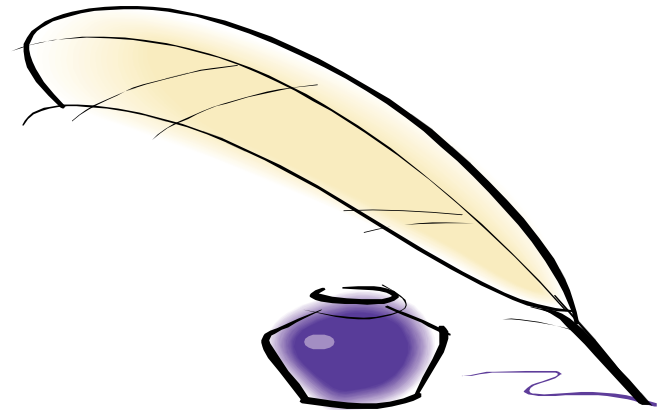
Dept. of Game Software

Notices

- 03/07: 502 → 501 등록 이동
- 03/21: HW 1 (Due: 03/28)

Plan

- MFC 화면 출력
 - Color point
 - CRect
 - Line
 - Text
 - Mapping Mode

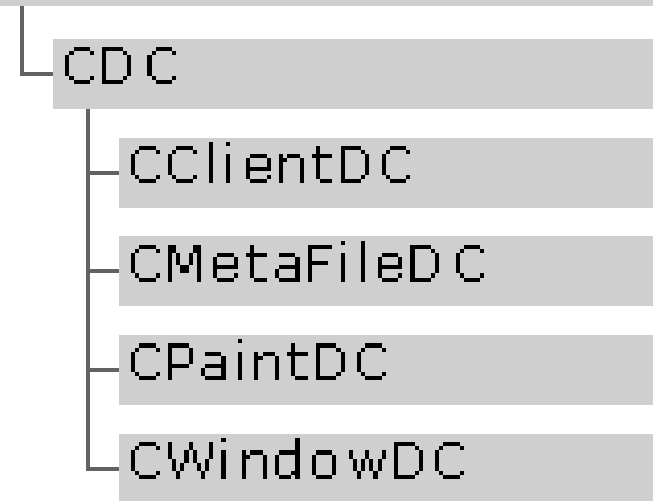


Windows CDC Class

- CDC class

Class	용도
CPaintDC	Client 영역에 출력할 때 (WM_PAINT message 처리기에서 만 사용)
CClientDC	Client 영역에 출력할 때 (WM_PAINT message 처리기를 제외한 다른 모든 곳에서 사용)
CWindowDC	Window의 전체 영역(Client 영역 + 비 client 영역)에 출력할 때
CMetaFileDC	메타 파일(Metafile)에 출력할 때

Object



Color Point

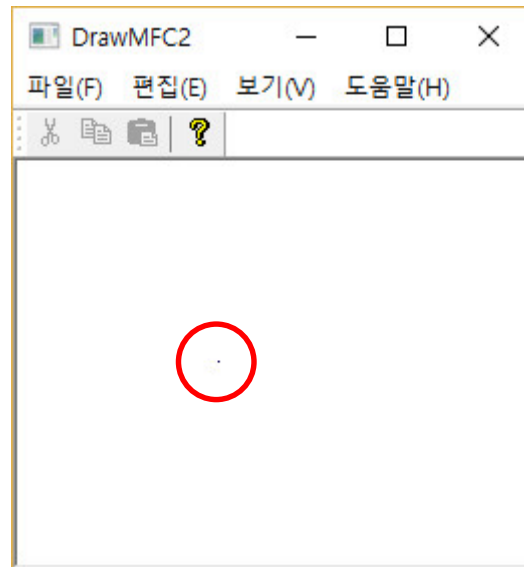
- Point 찍기 함수

이름	기능
GetPixel	화면의 특정 위치에 해당하는 점의 색을 얻는다.
SetPixel	화면의 특정 위치에 원하는 색의 점을 찍으며 원래의 점의 색을 리턴한다.
SetPixelV	SetPixel과 기능은 동일하지만 SetPixel 함수와 달리 원래의 점의 색을 리턴하지 않으므로 속도가 더 빠르다.

```
COLORREF color = dc.GetPixel (x,y);  
dc.SetPixelV(x, y, RGB(r, g, b));
```

Color Point

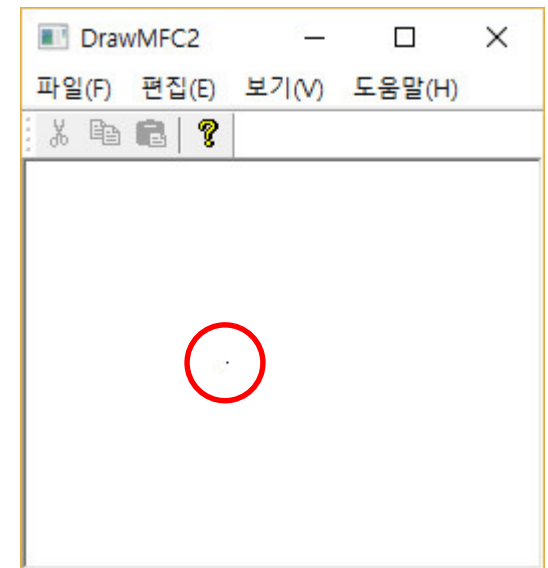
- 사용 예: 특정 위치에 point 찍기 (1/3)
 - SetPixel() 또는 SetPixelV() 사용
 - Blue color 값(COLORREF type) = 0x000000FF 사용



Color Point

- 사용 예: 특정 위치에 point 찍기 (2/3)
 - SetPixel() 또는 SetPixelV() 사용
 - Blue color 값(COLORREF type) = 0x000000FF 사용

```
// (100,100)위치에 파란색을 칠함  
dc.SetPixel(100, 100, 0x000000FF);  
// or  
dc.SetPixelV(100, 100, 0x000000FF);  
  
// (100,100)위치의 색을 가져옴  
COLORREF color = dc.GetPixel(100, 100);
```



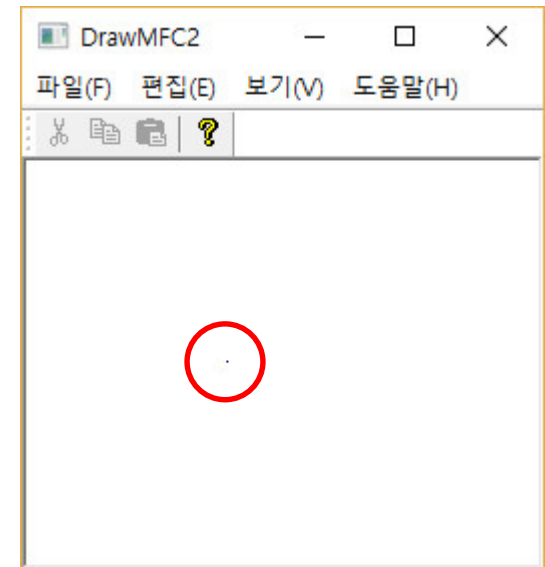
Color Point

- 사용 예: 특정 위치에 point 찍기 (3/3)
 - COLORREF: 색을 저장하는 변수 사용
 - 32 bit: 0x00rrggbb 형식
 - 여러가지 편한 macro 함수와 같이 사용: 예) RGB()

```
// RGB 매크로를 사용하여 COLORREF color 값으로 변환  
COLORREF color = RGB(r, g, b);
```

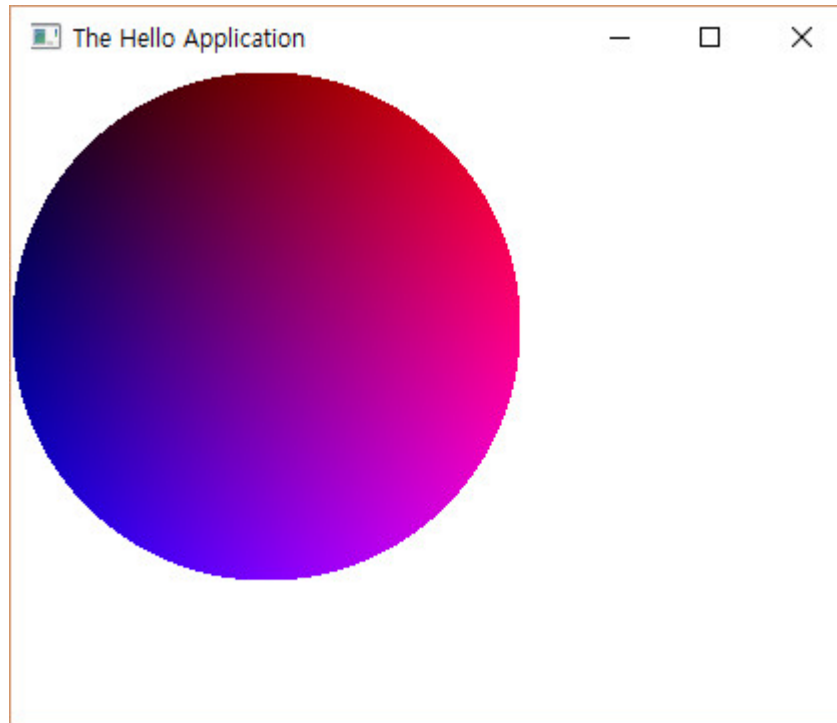
```
// 변수 r, g, b에 가져온 색의 R값, G값, B값을 저장  
int r = GetRValue(color);  
int g = GetGValue(color);  
int b = GetBValue(color);
```

```
// (100,100)위치에 파란색을 칠함  
dc.SetPixel(100, 100, RGB(0, 0, 100));
```



Color Point

- 사용 예: 서서히 색이 변하는 원 그리기 (1/2)
 - (x, y) 가 원의 내부 좌표일 때만 색을 칠함
 - 원의 좌표: $(x - r)(x - r) + (y - r)(y - r) < r * r$
 - $r = 255 / 2 = 127$



Color Point

- 사용 예: 서서히 색이 변하는 원 그리기 (2/2)
 - (x, y) 가 원의 내부 좌표일 때만 색을 칠함
 - 원의 좌표: $(x - r)(x - r) + (y - r)(y - r) < r * r$
 - $r = 255 / 2 = 127$

```
// 서서히 색이 변하는 원 그리기
for (int x = 0 ; x < 255; x++) {
    for (int y = 0; y < 255; y++) {
        //(x,y)가 원의 내부 좌표일 때만 색을 칠한다
        if ((x - 127) * (x - 127) + (y - 127) * (y - 127) < 127 * 127)
            dc.SetPixelV(x, y, RGB(x,0,y));
    }
}
```

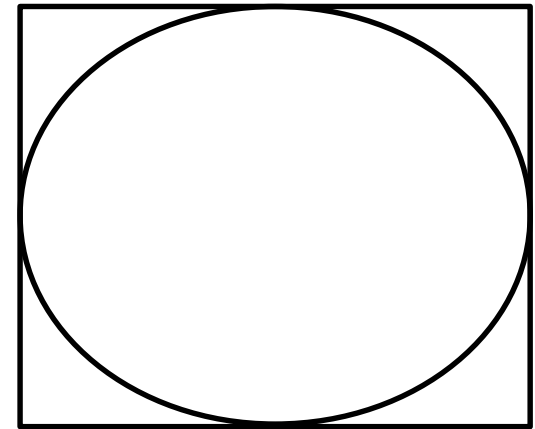
CRect

- 도형 그리기: 사각형과 원

이름	기능
Rectangle()	사각형을 그린다.
Ellipse()	사각형에 내접하는 타원을 그린다.

```
dc.Rectangle (x1, y1, x2, y2);  
dc.Ellipse (x1, y1, x2, y2);
```

(x1, y1)



(x2, y2)

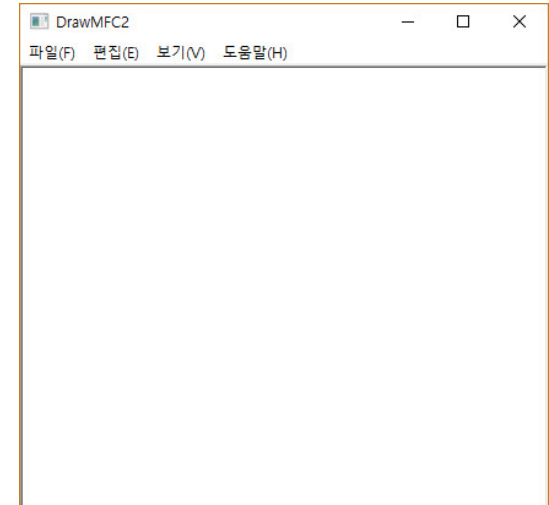
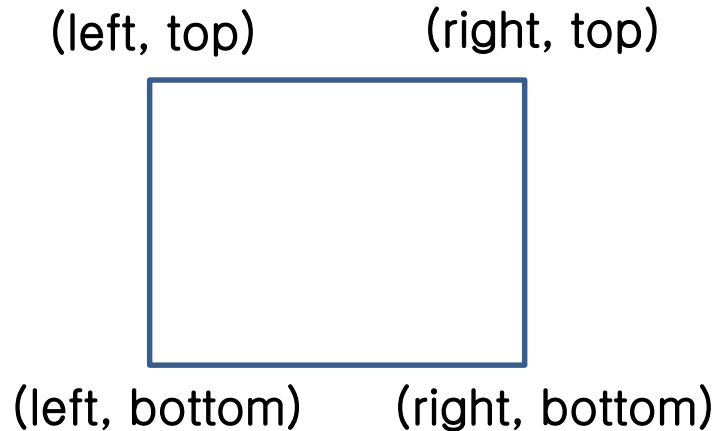
CRect

- Windows의 사용자 영역 size 구하기

```
CRect rect;  
GetClientRect(rect);
```

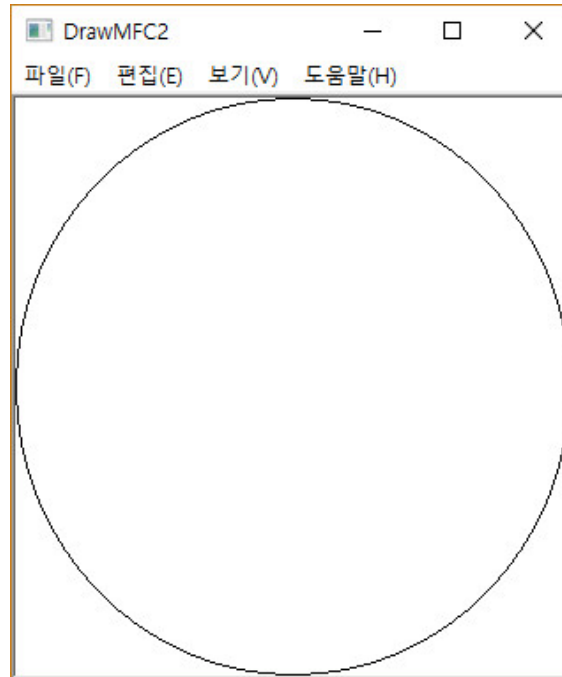
- CRect : 사각형 정보를 저장하는 class
 - member variables:

- bottom
- top
- right
- left



CRect

- 사용 예: Windows 사용자 영역에 꼭 차는 원 그리기 (1/2)
 - CRect 구조체 사용



CRect

- 사용 예: Windows 사용자 영역에 꼭 차는 원 그리기 (2/2)
 - CRect 구조체 사용

```
CRect rect;  
GetClientRect(rect);  
dc.Ellipse(rect);
```



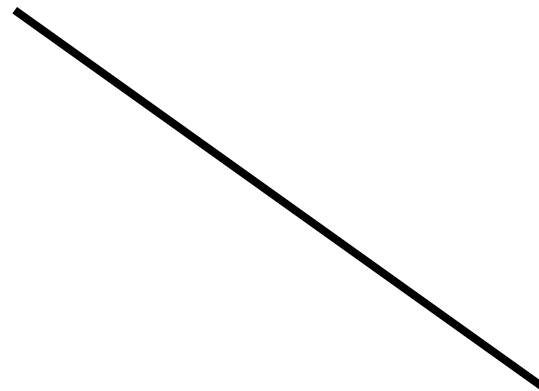
Line 그리기

- Line 그리기 함수

이름	기능
MoveTo()	현재 위치를 옮긴다.
LineTo()	현재 위치로부터 특정 위치까지 선을 그린 후 현재 위치를 갱신한다.

```
dc.MoveTo(x1,y1);  
dc.LineTo(x2,y2);
```

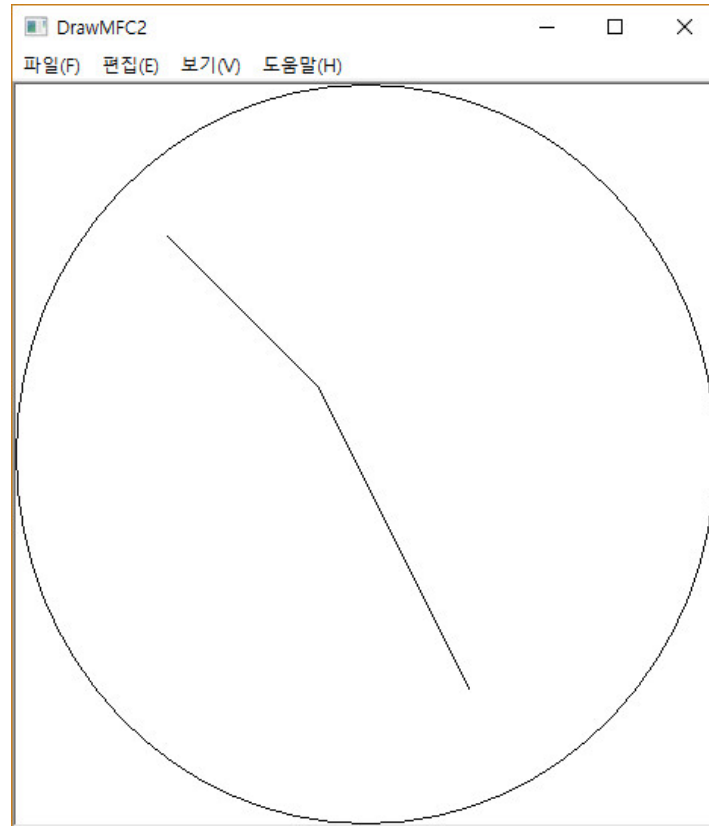
(x1,y1)



(x2,y2)

Line 그리기

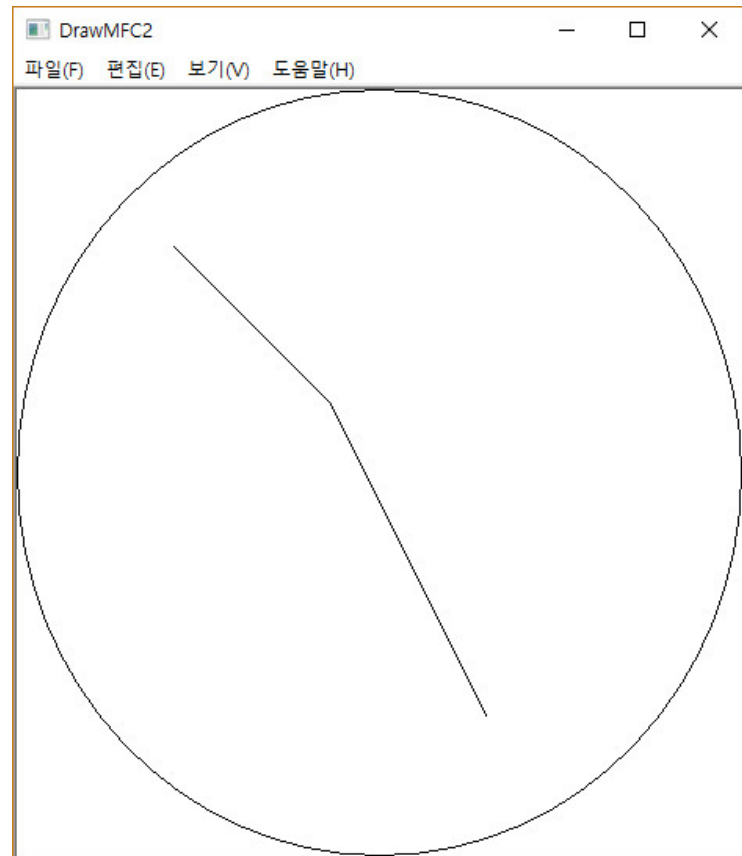
- 사용 예: 특정 points를 경유하는 line 그리기 (1/2)
 - Point 위치: (100, 100), (200, 200), (300, 400)



Line 그리기

- 사용 예 : 특정 points를 경유하는 line 그리기 (2/2)
 - Point 위치: (100, 100), (200, 200), (300, 400)

```
dc.MoveTo(100, 100);  
dc.LineTo(200, 200);  
dc.LineTo(300, 400);
```



Text 출력

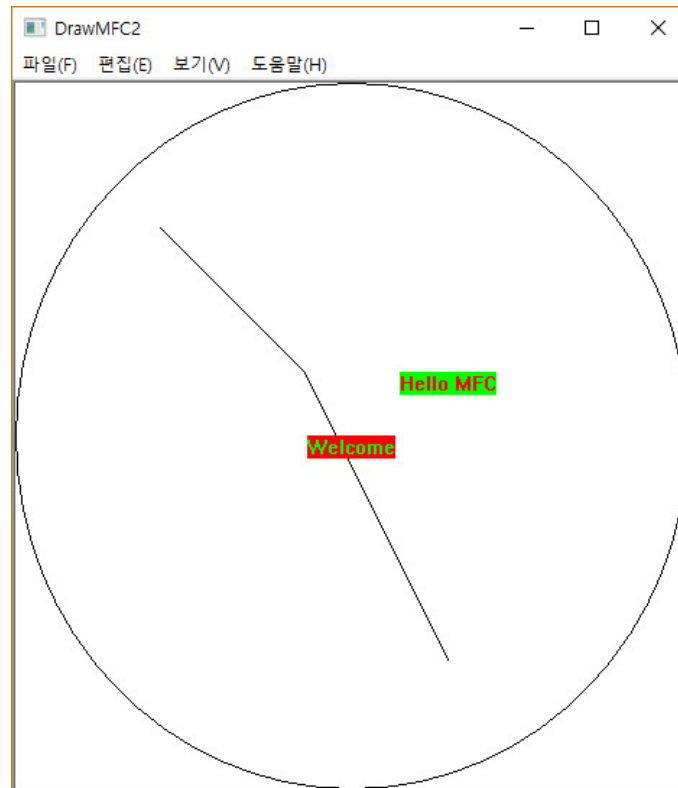
- Text 출력 함수

이름	기능
TextOut()	특정 위치에 문자열을 출력한다.
DrawText()	사각형을 기준으로 문자열을 출력한다.
SetTextColor()	문자의 색을 바꾼다.
SetBkColor()	문자의 배경색을 바꾼다.
SetTextAlign()	기준 위치에 대한 문자열의 정렬 방식을 정한다.

```
dc.SetTextColor(RGB(255, 0, 0));  
dc.SetBkColor(RGB(0, 255, 0));  
dc.SetTextAlign(TA_CENTER);  
dc.TextOut(300, 200, _T("Hello MFC"));
```

Text 출력

- 사용 예 : Client 영역 중간에 text 출력 (1/2)
 - CRect 사용



Text 출력

- 사용 예 : Client 영역 중간에 text 출력 (1/2)
 - CRect 사용

```
CRect rect;  
GetClientRect(rect);  
dc.SetTextColor(RGB(0, 255, 0));  
dc.SetBkColor(RGB(255, 0, 0));  
  
// dc.SetBkMode(TRANSPARENT);  
dc.SetTextAlign(TA_CENTER);  
dc.TextOut(rect.Width()/2, rect.Height()/2, _T("Welcome"));  
  
// or  
dc.DrawText(_T("Welcome"), &rect, DT_CENTER | DT_VCENTER | DT_SINGLELINE);
```

출력할 문자열 글자의 출력 맞춤 모양

문장을 출력할 범위를 의미하는 사각형

Text 출력

- Text 출력 속성 함수

속성	초기값	속성을 얻는 함수	속성을 변경하는 함수
텍스트 색상	검정색	GetTextColor()	SetTextColor()
배경 색상	흰색	GetBkColor()	SetBkColor()
배경 모드	OPAQUE	GetBkMode()	SetBkMode()
매핑 모드	MM_TEXT	GetMapMode()	SetMapMode()
그리기 모드	R2_COPYPEN	GetROP2()	SetROP2()
현재 위치	(0, 0)	GetCurrentPosition()	MoveTo()
펜	BLACK_PEN	SelectObject()	SelectObject()
브러시	WHITE_BRUSH	SelectObject()	SelectObject()
폰트	SYSTEM_FONT	SelectObject()	SelectObject()
비트맵	없음	SelectObject()	SelectObject()
팔레트	없음	SelectPalette()	SelectPalette()
영역	없음	SelectObject()	SelectObject()

Mapping Mode

- Windows 사용자 영역에서 2가지 길이 표시 방법
 - Device unit: Pixel
 - Logical unit: Real metric (mm, cm, inch, ...)

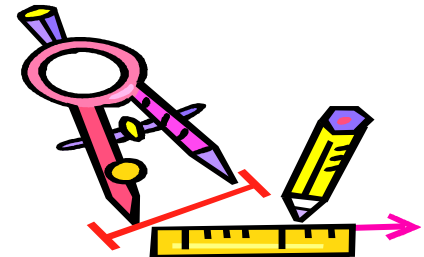
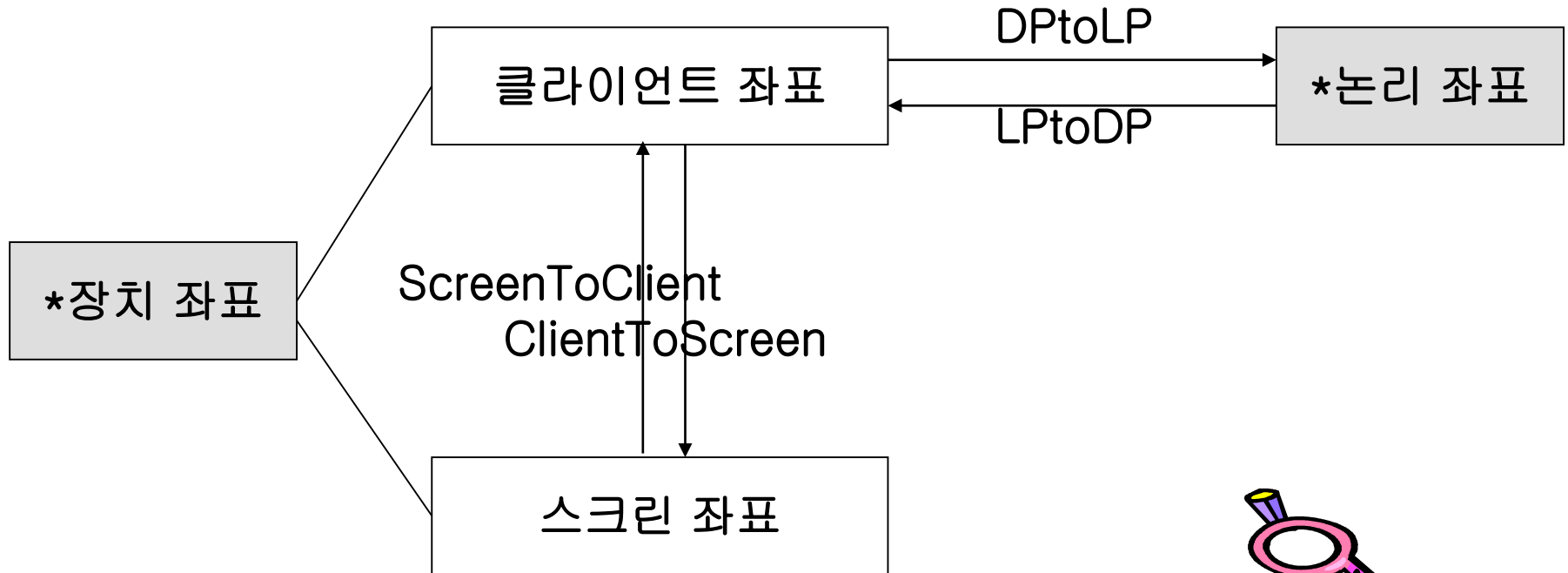
Mapping Mode	Unit(단위)	x축	y축
MM_TEXT	1 pixel	→ +x	↓ +y
MM_LOMETRIC	0.1 mm	→ +x	↓ -y
MM_HIMETRIC	0.01 mm	→ +x	↓ -y
MM_LOENGLISH	0.01 인치	→ +x	↓ -y
MM_HIENGLISH	0.001 인치	→ +x	↓ -y
MM_TWIPS	1/1440 인치	→ +x	↓ -y
MM_ISOTROPIC	사용자 정의(가로, 세로 길이 동일)	사용자 정의	사용자 정의
MM_ANISOTROPIC	사용자 정의	사용자 정의	사용자 정의

```
dc.SetMapMode(MM_LOMETRIC);  
dc.Rectangle(0, 0, 100, -100); // 길이가 10mm짜리 정사각형 출력
```

Mapping Mode

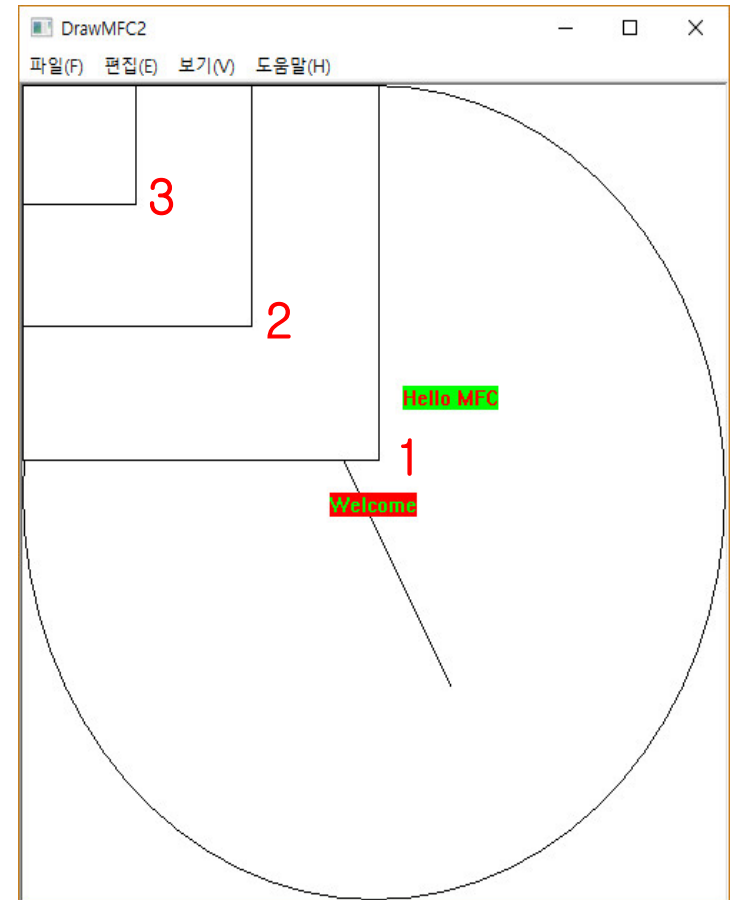
- 좌표 변환 함수

- 논리 좌표: Programmer가 사용하는 단위
- 장치 좌표: 운영체제 내부적으로 사용하는 단위 (pixel)



Mapping Mode

- 사용 예: 크기가 다른 삼각형 그리기 (1/2)
 - 1번: 250 by 250 pixels
 - 2번: 5 by 5 cm
 - 3번: 2.5 by 2.5 cm를 pixel로 출력



Mapping Mode

- 사용 예: 크기가 다른 삼각형 그리기 (1/2)
 - 1번: 250 by 250 pixels
 - 2번: 5 by 5 cm
 - 3번: 2.5 by 2.5 cm를 pixel로 출력

```
// 1번 사각형
```

```
dc.Rectangle(0, 0, 250, 250);
```

```
// 2번 사각형
```

```
dc.SetMapMode(MM_LOMETRIC);
```

```
dc.Rectangle(0, 0, 500, -500);
```

```
// 3번 사각형
```

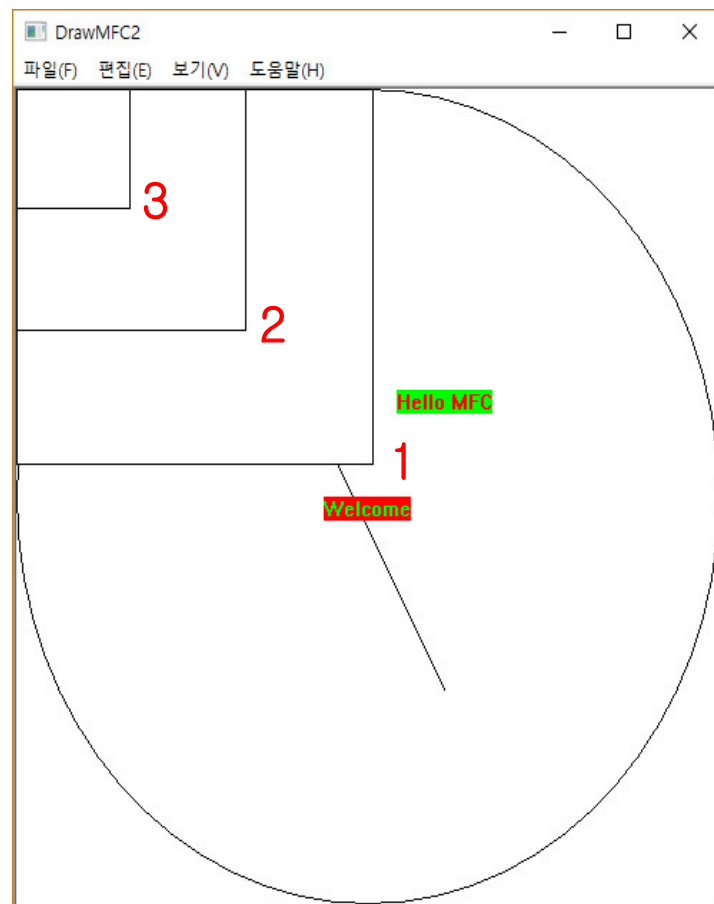
```
POINT p;
```

```
p.x = 500; p.y = -500;
```

```
dc.LPtoDP(&p);
```

```
dc.SetMapMode(MM_TEXT);
```

```
dc.Rectangle(0, 0, p.x/2, p.y/2);
```



Q & A